

A time-delayed observer for fault detection and isolation in industrial robots

F. Caccavale*, P. Chiacchio† and I. D. Walker‡

(Received in Final Form: November 20, 2005. First published online: February 14, 2006)

SUMMARY

In this paper a discrete-time observer-based approach to Fault Detection and Isolation (FDI) for industrial robotic manipulators is presented and experimentally tested. In order to counteract the effects of unmodeled dynamics and disturbances, a time-delayed estimate of such effects is adopted. Remarkably, the observer is designed directly in the discrete-time domain. The performance of the proposed approach are experimentally verified on a six-degrees-of-freedom industrial robot.

KEYWORDS: Robots; Observers; Fault detection; Fault isolation.

I. INTRODUCTION

The adoption of effective fault detection and isolation (FDI) techniques is becoming critical to ensure higher levels of safety and productivity in automated plants and autonomous systems.

Hence, considerable research efforts have been spent to seek for systematic approaches to fault detection and isolation in dynamical systems. In this framework special attention has been paid to robotic systems, especially for those operating in remote or hazardous environments, where a high degree of safety and self-diagnostics capabilities are required. However, the development and the application of FDI techniques is also of the utmost importance for industrial robots, where the main objectives are the achievement of a safe man-machine interaction as well as a quick and appropriate reaction of the system to the occurrence of failures.

The main goal of an FDI algorithm is the monitoring of the system during its normal working conditions so as to detect the occurrence of failures (*fault detection*), and recognize the location and the size of the failures (*fault isolation*). In the model-based approach to FDI, this goal is achieved by comparing the actual system's behaviour with the corresponding expected behaviour derived via its

mathematical model. Usually, the output of a fault detection algorithm is a set of variables sensitive to the occurrence of a failure (*residuals*). Namely, when a failure occurs, a *fault signature* affects the residuals. Then, the information from the signatures is processed to identify the size and the location of the fault. The interested reader is referred to references [1, 2] for a wide overview of the existing model-based FDI techniques. As for the case of nonlinear dynamical systems the fault detection methods can be roughly regrouped in three main classes: observer-based approaches,¹ parameter estimation techniques^{3,4} and algorithms based on learning methodologies.^{5–9}

Several FDI techniques for mechanical manipulators have been developed in the literature, based on parameter estimation,^{4,10} on the combined use of state observers and fuzzy logic residuals evaluation¹¹ or on nonlinear adaptive state estimation.¹² In reference [13] an analytical redundancy concept¹⁴ is applied together with an adaptive update of the thresholds on the residuals, while in reference [15] a neural network is employed to match an unknown fault. More recently, a computationally cheap approach, based on the so-called generalized momenta, has been proposed in reference [16] and recently extended in reference [17] to take into account parametric uncertainties in the robot model.

This paper focuses on nonlinear observer-based FDI approaches for mechanical manipulators. The goal is to develop an FDI scheme suitable for the application to conventional industrial manipulators, which overcomes the typical problems of industrial setups, i.e. lack of knowledge of some terms in the mathematical model, effects of sensory noise and of discrete-time implementation.

Usually, the observer-based methods require a model of the system to be operated in parallel to the process (*diagnostic observer*) in an open-loop fashion.¹¹ Then, the residuals are computed as the difference between the measured output variables and those predicted via the diagnostic observer. Assuming an exact knowledge of the manipulator dynamics, the residuals should become nonzero when a fault occurs. However, perfect knowledge of the manipulator model is rarely a reasonable assumption, especially for industrial gear-driven robots, where effects like backlash and friction become not negligible, and yet difficult to model. Further, the diagnostic observer has to be implemented as a discrete-time system eventually working at a low sampling rate. In order to limit the computational burden, a simple discretization scheme should be used to implement the observer; this may lead to a drifting behaviour of the residuals. Hence, the observer cannot be operated in an open-loop fashion: some information coming from the real model (through the

* Corresponding author. Dipartimento di Ingegneria e Fisica dell'Ambiente, Università degli Studi della Basilicata Via dell'Ateneo Lucano 10, 85100 Potenza (Italy) E-mail: caccavale@unibas.it

†Dipartimento di Ingegneria dell'Informazione ed Ingegneria Elettrica, Università degli Studi di Salerno Via Ponte Don Melillo, 84084 Fisciano, Salerno (Italy).

‡Department of Electrical and Computer Engineering Clemson, University Fluor Daniel Engineering Innovation Building Clemson, SC 29634 (USA).

sensors) must be provided to the observer in order to improve its robustness to both model uncertainties and discretization errors. However, it turns out (since faults in the sensors propagate into the observer) that good robustness properties can be achieved at the expense of a reduced sensitivity to failures; this requires that solutions based on a trade-off between these two performance indexes must be devised.

Hence, the design of a diagnostic observer must take into account two main requirements:

- In order to take into account the unavoidable effects of the discretization, the diagnostic observer should be designed directly in the discrete-time domain.
- Information provided by sensor measurements must be used, so as to improve the robustness of the observer without a destructive impact on the sensitivity to the failures.

In order to fulfill the above requirements, a novel diagnostic observer is proposed in this paper. The observer is based on a linear feedback of the observation error and a nonlinear time-delayed term providing an estimate of the unknown dynamics and errors due to discretization. A suitable choice of residuals, computed on the basis of the observer outputs, allows the design of a simple and reliable fault isolation procedure, i.e. the type and the size of the failures can be determined from the signatures detected from the residuals. Preliminary studies on this approach, including experimental results, are reported in references [18, 19].

A convergence analysis of the observation error is carried out and the sensitivity of the resulting FDI scheme to the faults is discussed. Finally, the proposed FDI scheme is experimentally tested on a six-degree-of-freedom industrial manipulator with open control architecture. The results show that the proposed scheme is able to successfully counteract both the effect of uncertain model knowledge and of discretization. In fact, low residuals in the fault free case are obtained, while clear and distinct fault signatures arise in the case of both sensor and actuator failures.

II. MODELING

II.1. Equations of motion

The dynamic model of an n -degrees-of-freedom rigid robot in the continuous time can be written in the form

$$\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{n}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) = \boldsymbol{\tau}(t) - \boldsymbol{\delta}(t), \quad (1)$$

where \mathbf{q} and $\boldsymbol{\tau}$ denote the $(n \times 1)$ vectors of joint variables and joint torques, respectively, and $\boldsymbol{\delta}$ represents the term collecting all the unmodeled dynamic terms and disturbances. In (1) \mathbf{M} is the $(n \times n)$ symmetric and positive definite inertia matrix and

$$\mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{F}\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (2)$$

is the $(n \times 1)$ vector collecting the Coriolis and centrifugal $(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}})$, friction $(\mathbf{F}\dot{\mathbf{q}})$ and gravitational $(\mathbf{g}(\mathbf{q}))$ terms. Hereafter, it is assumed that only an approximate model of the manipulator is known, i.e. only nominal estimates (i.e.

$\widehat{\mathbf{M}}(\mathbf{q}), \widehat{\mathbf{n}}(\mathbf{q}, \dot{\mathbf{q}})$) are available for the terms in (1), while $\boldsymbol{\delta}$ shall be considered as an exogenous unknown input.

A choice for the state variables of the system is represented by the $(2n \times 1)$ vector

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \end{bmatrix} = \begin{bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{bmatrix}. \quad (3)$$

The state-space equations of the manipulator are then given by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{h}(\mathbf{x}(t)) + \mathbf{B}(\mathbf{x}(t))\mathbf{u}(t) + \boldsymbol{\eta}(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \end{cases} \quad (4)$$

where \mathbf{y} denotes the $(p \times 1)$ output vector, $\mathbf{u} = \boldsymbol{\tau}$ and \mathbf{C} is the $(p \times 2n)$ output matrix. The matrices \mathbf{A} and \mathbf{B} in (4) are defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{O}_n & \mathbf{I}_n \\ \mathbf{O}_n & \mathbf{O}_n \end{bmatrix}, \quad \mathbf{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{O}_n \\ \widehat{\mathbf{M}}^{-1}(\mathbf{x}_1) \end{bmatrix}, \quad (5)$$

where \mathbf{O}_n denotes the $(n \times n)$ null matrix and \mathbf{I}_n denotes the $(n \times n)$ identity matrix. If the whole state (i.e. joints positions and velocities) is measurable, then $p = 2n$ and $\mathbf{C} = \mathbf{I}_{2n}$.

The other two terms in (4) are given by

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_n \\ -\widehat{\mathbf{M}}^{-1}(\mathbf{x}_1)\widehat{\mathbf{n}}(\mathbf{x}_1, \mathbf{x}_2) \end{bmatrix}, \quad (6)$$

and

$$\begin{aligned} \boldsymbol{\eta}(\mathbf{x}, \mathbf{u}) &= \begin{bmatrix} \mathbf{0}_n \\ -\widehat{\mathbf{M}}^{-1}(\mathbf{x}_1)(\widetilde{\mathbf{M}}(\mathbf{x}_1)\dot{\mathbf{x}}_2 + \widetilde{\mathbf{n}}(\mathbf{x}_1, \mathbf{x}_2) + \boldsymbol{\delta}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_n \\ -(\widehat{\mathbf{M}}^{-1}(\mathbf{x}_1) - \mathbf{M}^{-1}(\mathbf{x}_1))(\mathbf{u} - \mathbf{n}(\mathbf{x}_1, \mathbf{x}_2) - \boldsymbol{\delta}) \\ -\widehat{\mathbf{M}}^{-1}(\mathbf{x}_1)(\widetilde{\mathbf{n}}(\mathbf{x}_1, \mathbf{x}_2) + \boldsymbol{\delta}) \end{bmatrix}, \end{aligned} \quad (7)$$

where the notation ‘ \sim ’ is used to indicate the difference between true and nominal quantities (e.g., $\widetilde{\mathbf{M}} = \mathbf{M} - \widehat{\mathbf{M}}$) and $\mathbf{0}_n$ denotes the $(n \times 1)$ null vector. Hence, the term $\boldsymbol{\eta}$ takes into account model uncertainties and disturbances.

Throughout the paper, it is assumed that the measurements of the output variables are available at a sampling time T , and the input torques are maintained constant over each time interval $\mathcal{I}_k \equiv [kT, (k+1)T]$, where $k \geq 0$ is an integer. Thus, it is worth looking for a discrete-time equivalent of the model (4). To the purpose, consider a Taylor expansion of $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ at $t_{k+1} = (k+1)T$, with starting point $t_k = kT$

$$\begin{cases} \mathbf{x}_1(k+1) = \mathbf{x}_1(k) + T\mathbf{x}_2(k) + \boldsymbol{\rho}_{d1}(k) \\ \mathbf{x}_2(k+1) = \mathbf{x}_2(k) + T\dot{\mathbf{x}}_2(k) + \boldsymbol{\rho}_{d2}(k), \end{cases} \quad (8)$$

where the discrete time variables t_k and t_{k+1} have been denoted simply by the integers k and $k+1$, respectively. In (8) the two quantities $\boldsymbol{\rho}_{d1}$ and $\boldsymbol{\rho}_{d2}$ represent the local discretization errors due to the truncation of the series; they depend linearly on the joint accelerations and jerks,

respectively, evaluated at suitable intermediate points in \mathcal{I}_k . Thus, a simple discrete-time model based on the first order Euler method, can be found

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}_d \mathbf{x}(k) + \mathbf{h}_d(\mathbf{x}(k)) + \mathbf{B}_d(\mathbf{x}(k))\mathbf{u}(k) + \boldsymbol{\eta}_d(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k), \end{cases} \quad (9)$$

where

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{I}_n & T\mathbf{I}_n \\ \mathbf{O}_n & \mathbf{I}_n \end{bmatrix}, \quad \boldsymbol{\eta}_d = T\boldsymbol{\eta} + \begin{bmatrix} \boldsymbol{\rho}_{d1} \\ \boldsymbol{\rho}_{d2} \end{bmatrix}, \quad (10)$$

and

$$\mathbf{B}_d = T\mathbf{B}, \quad \mathbf{h}_d = T\mathbf{h}. \quad (11)$$

The features of the Euler method for numerical integration of differential equations are well-known (see, e.g., reference [20]): the *local* truncation errors are given by $\boldsymbol{\rho}_{d1}$ and $\boldsymbol{\rho}_{d2}$, while the *global* accumulated error is of the same order of magnitude as T . It is worth noticing that different discrete dynamic models have been proposed in the literature, ensuring better performance both for simulations and control purposes (see, e.g. the model proposed in reference [21]). However, these models are not so simple as the Euler-based one, and they are not tailored for *direct dynamics* purposes (i.e. for the evaluation of the manipulator state at the current step, given the state and the input at the previous step).

If velocity measurements are not available the joint velocities are usually obtained via numerical reconstruction, e.g. using a first order difference of the measured joint positions

$$\mathbf{x}_{2n}(k) = \frac{\mathbf{x}_1(k) - \mathbf{x}_1(k-1)}{T}, \quad (12)$$

where the relationship with the true time derivative can be expressed as $\mathbf{x}_2(k) = \mathbf{x}_{2n}(k) + \boldsymbol{\rho}_{2n}(k)$, being $\boldsymbol{\rho}_{2n}(k)$ an error term depending on the chosen discretization method (e.g. the first-order backward difference as in (12)).

In this case the model (9) can be rewritten in terms of the numerically reconstructed velocities rather than the actual velocities, and a new discrete-time model can be derived, which is formally similar to (9)

$$\begin{cases} \mathbf{z}(k+1) = \mathbf{A}_d \mathbf{z}(k) + \mathbf{h}_d(\mathbf{z}(k)) + \mathbf{B}_d(\mathbf{z}(k))\mathbf{u}(k) + \boldsymbol{\eta}_z(k) \\ \mathbf{y}_z(k) = \mathbf{z}(k), \end{cases} \quad (13)$$

where

$$\mathbf{z}(k) = \begin{bmatrix} \mathbf{x}_1(k) \\ \mathbf{x}_{2n}(k) \end{bmatrix}, \quad (14)$$

and $\boldsymbol{\eta}_z$ is a suitably defined vector depending on $\boldsymbol{\eta}_d$ and $\boldsymbol{\rho}_{2n}$. Therefore, the approach described in the following for the design of the diagnostic observer can be extended to this case. Clearly, the new model is written in terms of a new set of state variables, in which the actual joint velocities are replaced with their numerical counterparts; this is not to be considered as a drawback, since the main goal of an FDI scheme is represented by failure detection and not by the state estimation.

II.2. Fault modeling

The classes of failures considered in this work are those of *actuator* faults and *sensor* faults. The former can be defined as the class of failures occurring in the joint actuating systems, i.e. the systems composed of the driving motors and, eventually, the corresponding gear trains. This class of failures can be represented as a discrepancy between the nominal (commanded) and the true driving torques acting at the robot joints

$$\boldsymbol{\tau}_{true}(t) = \boldsymbol{\tau}_{nom}(t) + \boldsymbol{\delta}\boldsymbol{\tau}(t), \quad (15)$$

where $\boldsymbol{\delta}\boldsymbol{\tau}$ represents the unknown fault. Hence, an actuator fault occurring at $t = t_k$ results in an effective control input given by

$$\mathbf{u}(k) = \bar{\mathbf{u}}(k) + \boldsymbol{\delta}\mathbf{u}(k), \quad (16)$$

where $\bar{\mathbf{u}}$ is the nominal torque input.

Usually, only joint position sensors are available in industrial robotic set-ups; hence only position sensor failures will be considered. A sensor fault occurs when the difference between the sensor readings and the true values of the measured joint variable exceeds a preassigned value, i.e.

$$\mathbf{q}_{meas}(t) = \mathbf{q}_{true}(t) + \boldsymbol{\delta}\mathbf{q}(t), \quad (17)$$

where $\boldsymbol{\delta}\mathbf{q}$ represents the unknown fault. Hence, a sensor fault occurring at $t = t_k$ can be modeled as

$$\bar{\mathbf{x}}_1(k) = \mathbf{x}_1(k) + \boldsymbol{\delta}\mathbf{x}_1(k), \quad (18)$$

where $\bar{\mathbf{x}}_1$ represents the output of the sensor and \mathbf{x}_1 the true value of the joint coordinates.

III. OBSERVER-BASED FAULT DETECTION AND ISOLATION

In this section the observer for fault detection proposed in reference [18] is extended and analyzed. The main goal is to design a diagnostic observer able to track the behaviour of the manipulator despite the discretization errors and uncertainties in the available mathematical model. In the following, the proposed FDI scheme is developed with reference to the discrete-time model (9), the derivation of the results being formally similar for the model in the form (13). However, the specific features coming from the use of the model (13) will be noted when needed.

III.1. Diagnostic observer

Let us consider the manipulator operating in the absence of failures. Taking into account the discrete-time manipulator model (9) and assuming the whole state measurable (i.e. $\mathbf{C} = \mathbf{I}_{2n}$), the proposed observer is described by the following equations

$$\begin{aligned} \hat{\mathbf{x}}(k+1) &= \mathbf{A}_d \hat{\mathbf{x}}(k) + \mathbf{h}_d(\hat{\mathbf{x}}(k)) + \mathbf{B}_d(\hat{\mathbf{x}}(k))\mathbf{u}(k) \\ &\quad + \mathbf{K}_o \boldsymbol{\varepsilon}(k) + \hat{\boldsymbol{\eta}}_d(k), \end{aligned} \quad (19)$$

where $\boldsymbol{\varepsilon} = \mathbf{x} - \hat{\mathbf{x}}$ is the observation error, and the notation ' $\hat{\cdot}$ ' denotes the estimated variables. In (19) the term $\mathbf{K}_o \boldsymbol{\varepsilon}$ performs a feedback action based on the observation error

$$\mathbf{K}_o \boldsymbol{\varepsilon} = \begin{bmatrix} \mathbf{K}_1 & T \mathbf{I}_n \\ \mathbf{O}_n & \mathbf{K}_2 \end{bmatrix} \boldsymbol{\varepsilon}, \quad (20)$$

where \mathbf{K}_1 and \mathbf{K}_2 are positive definite diagonal ($n \times n$) matrices. In order to improve the robustness of the observer, a nonlinear delayed disturbance compensation term is provided by

$$\hat{\boldsymbol{\eta}}_d(k) = \mathbf{x}(k) - [\mathbf{A}_d \mathbf{x}(k-1) + \mathbf{h}_d(\mathbf{x}(k-1)) + \mathbf{B}_d(\mathbf{x}(k-1))\mathbf{u}(k-1)]. \quad (21)$$

Notice that the computation of (21) requires only quantities available from the previous step and measurements at the current step, and thus the computational burden is kept limited. It can be recognized that the following equality holds

$$\hat{\boldsymbol{\eta}}_d(k) = \boldsymbol{\eta}_d(k-1), \quad (22)$$

which means that $\hat{\boldsymbol{\eta}}_d$ can be considered a good estimate of the effects due to discretization errors and unknown dynamics if $\boldsymbol{\eta}_d(k) \approx \boldsymbol{\eta}_d(k-1)$, i.e. within the adopted sampling time $\boldsymbol{\eta}_d$ is subject to negligible variations. Hence, it is assumed that the following condition holds

$$\|\Delta \boldsymbol{\eta}_d(k)\| = \|\boldsymbol{\eta}_d(k) - \boldsymbol{\eta}_d(k-1)\| \leq \alpha, \quad \forall k > 0, \quad (23)$$

where α is a positive constant and Δ denotes the first-order difference operator. It can be shown that the bound α in (23) depends on the manipulator joint trajectories (i.e. increases with joint velocities, accelerations and jerks), the adopted sampling time, and the degree of uncertainty on the model knowledge. Of course, an estimate of such a bound (e.g. obtained via mathematical model) would lead to extremely conservative results; it is expected that a more refined analysis would lead to time-varying bounds dependent on the actual and past states of the system. However, such an analysis is actually beyond the scope and the goals of this paper and will be omitted. For the sake of completeness, the time history of the values of $\|\Delta \boldsymbol{\eta}_{d1}(k)\|$ (i.e. the norm of the first n components) and $\|\Delta \boldsymbol{\eta}_{d2}(k)\|$ (i.e. the norm of the last n components) recorded during a typical robotic task has been reported in Fig. 1 for a sampling time of $T = 5$ ms.

Notice that the above strategy is based on the same ideas as the well-known time delay feedback control technique,²² which has proven to be a simple and effective strategy for controlling uncertain systems; here the same ideas are applied to an estimation problem rather than to a control task.

The resulting error equation takes on the following form

$$\boldsymbol{\varepsilon}(k+1) = \mathbf{F} \boldsymbol{\varepsilon}(k) + \Delta \boldsymbol{\eta}_d(k), \quad (24)$$

where $\mathbf{F}_i = \mathbf{I}_n - \mathbf{K}_i$ ($i = 1, 2$) and $\mathbf{F} = \text{block diag}\{\mathbf{F}_1, \mathbf{F}_2\}$ is diagonal.

In order to investigate the convergence of the observation error, consider the solution of the difference equation (24)

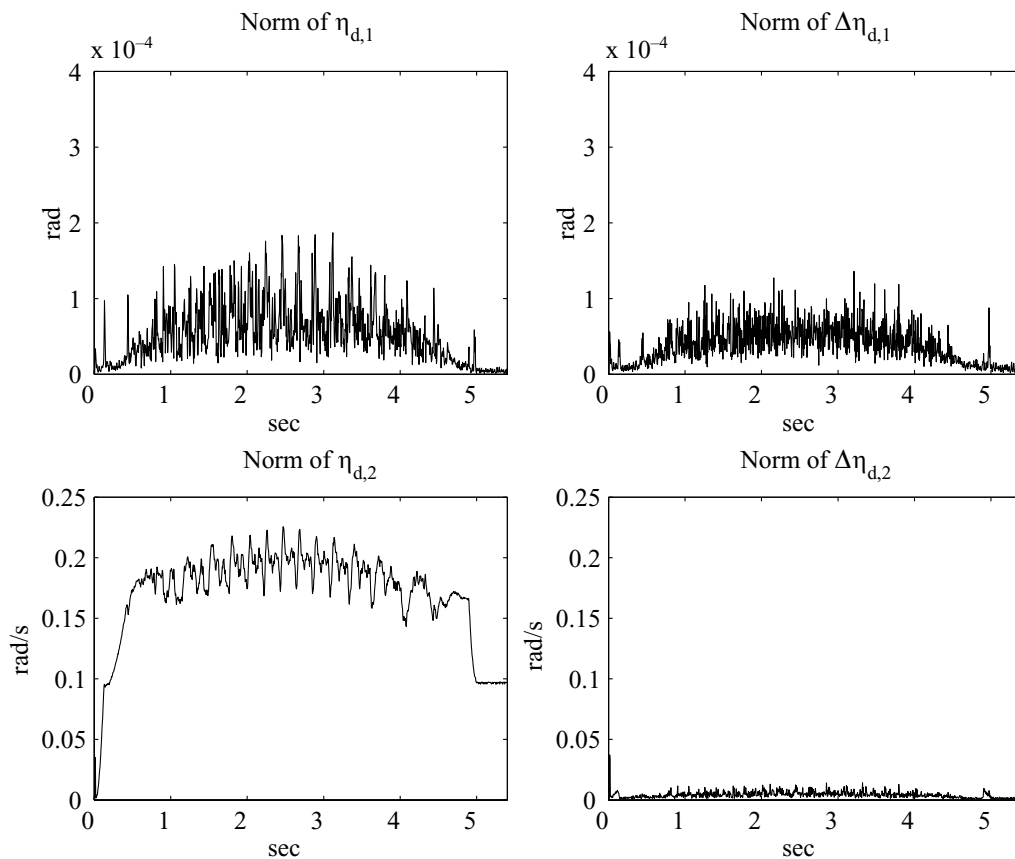


Fig. 1. Norm of $\boldsymbol{\eta}_d(k)$ for a typical robotic task (left) – Norm of $\Delta \boldsymbol{\eta}_d(k)$ for a typical robotic task (right). Sampling time: $T = 5$ ms.

obtained via its recursive application

$$\boldsymbol{\varepsilon}(l) = \mathbf{F}^l \boldsymbol{\varepsilon}(0) + \sum_{j=1}^l \mathbf{F}^{l-j} \boldsymbol{\Delta} \boldsymbol{\eta}_d(j-1), \quad (25)$$

with $l \geq 0$. A bound on $\|\boldsymbol{\varepsilon}\|$ can be easily derived from (23) and (25) as

$$\|\boldsymbol{\varepsilon}(l)\| \leq F^l \|\boldsymbol{\varepsilon}(0)\| + \alpha \sum_{j=0}^{l-1} F^j, \quad (26)$$

where $F = \|\mathbf{F}\|$; notice that both the terms on the right-hand side of (26) are bounded for all finite times. Assume that the feedback gain in (19) is chosen such that $0 < k_i < 2$, which implies $0 < F < 1$; then $\|\boldsymbol{\varepsilon}\|$ is asymptotically bounded as

$$\|\boldsymbol{\varepsilon}(\infty)\| \leq \alpha \sum_{j=0}^{\infty} F^j = \frac{\alpha}{1-F}. \quad (27)$$

Thus, the error norm is asymptotically bounded in a ball whose radius depends on the size of α and the chosen gains. Notice that the bound on the error norm decreases as $F \rightarrow 0$, i.e. for $F \rightarrow 0$ the bound tends to α . Hence, the observation error does not drift away from the actual system output, as expected if a discrete-time observer is operated in open loop.

It is worth remarking that the nonlinear disturbance compensation term could be computed on the basis of the measurements at the actual (i.e. $(k+1)$ -th) step

$$\widehat{\boldsymbol{\eta}}_d(k) = \mathbf{x}(k+1) - [\mathbf{A}_d \mathbf{x}(k) + \mathbf{h}_d(\mathbf{x}(k)) + \mathbf{B}_d(\mathbf{x}(k))\mathbf{u}(k)], \quad (28)$$

which leads to $\widehat{\boldsymbol{\eta}}_d(k) = \boldsymbol{\eta}_d(k)$. Hence, the error term $\boldsymbol{\Delta} \boldsymbol{\eta}_d$ in the resulting error equation disappears. This approach can be successfully pursued, as in reference [23], for designing state observers for uncertain dynamic systems, but it is not acceptable for fault detection purposes. In fact, as discussed in the following, $\boldsymbol{\eta}_d(k)$ carries information about the fault itself; hence, the complete compensation disturbance term as in (28) leads to destroying the signature of the residuals.

III.2. Residuals generation

If the diagnostic observer (19)–(21) is used, then the vector of residuals could be chosen simply equal to the observation error $\mathbf{r}_\varepsilon(k) = \boldsymbol{\varepsilon}(k)$. In the case of a sensor fault occurring at $t_{fault} = k_f T$ the residual evaluated at the same step becomes

$$\mathbf{r}_\varepsilon(k_f) = \mathbf{F} \boldsymbol{\varepsilon}(k_f - 1) + \begin{bmatrix} \delta \mathbf{x}_1(k_f) \\ \mathbf{0}_n \end{bmatrix} + \boldsymbol{\Delta} \boldsymbol{\eta}_d(k_f - 1). \quad (29)$$

On the other hand, if an actuator fault occurs at $t_{fault} = k_f T$ the residual evaluated one step later becomes

$$\mathbf{r}_\varepsilon(k_f + 1) = \mathbf{F} \boldsymbol{\varepsilon}(k_f) + \mathbf{B}_d(\mathbf{x}(k_f)) \delta \mathbf{u}(k_f) + \boldsymbol{\Delta} \boldsymbol{\eta}_d(k_f). \quad (30)$$

Hence, for fault isolation purposes (i.e. determination of the size and the location of the fault) it could be desirable to perform a decoupling transformation, such that the fault size can be detected without any configuration dependent amplitude factor (i.e. \mathbf{B}_d) and without the contribution of the past values of the error (i.e. $\mathbf{F} \boldsymbol{\varepsilon}$). Therefore, the fault isolation can be performed by examining the new residual vector

$$\mathbf{r}(k) = \mathbf{B}_d^*(\mathbf{x}(k-1))[\boldsymbol{\varepsilon}(k) - \mathbf{F} \boldsymbol{\varepsilon}(k-1)], \quad (31)$$

where

$$\mathbf{B}_d^*(\mathbf{x}) = \begin{bmatrix} \mathbf{I}_n & \mathbf{O}_n \\ \mathbf{O}_n & T^{-1} \widehat{\mathbf{M}}(\mathbf{x}_1) \end{bmatrix}. \quad (32)$$

If a fault affects the position sensor readings at $t_{fault} = k_f T$

$$\mathbf{r}(k_f) = \begin{bmatrix} \delta \mathbf{x}_1(k_f) \\ \mathbf{0}_n \end{bmatrix} + \mathbf{B}_d^*(\mathbf{x}(k_f - 1)) \boldsymbol{\Delta} \boldsymbol{\eta}_d(k_f - 1), \quad (33)$$

while an actuator fault occurring at the same step $t_{fault} = k_f T$ affects the residual as follows

$$\mathbf{r}(k_f + 1) = \delta \mathbf{u}(k_f) + \mathbf{B}_d^*(\mathbf{x}(k_f)) \boldsymbol{\Delta} \boldsymbol{\eta}_d(k_f). \quad (34)$$

It can be easily recognized that a sensor fault occurring at $k = k_f$ affects the first n components of $\mathbf{r}(k_f)$, while an actuator fault occurring at $k = k_f$ affects the last n components of $\mathbf{r}(k_f + 1)$. This shows that the fault signatures on the residuals vector possess different characteristics for the two classes of failures considered.

If joint velocities are obtained from measured joint via numerical differentiation, an improvement of detection capabilities for sensor faults is obtained. In fact, the reconstructed velocity signal can be seen as output by a “virtual” velocity sensor providing \mathbf{x}_{2n} , i.e. second component of the state vector $\mathbf{z}(k)$. It can be recognized that the occurrence of a sensor fault at $t_{fault} = k_f T$ produces a signature also on the last n components of the residual vector, i.e. the components corresponding to actuator fault signatures:

$$\mathbf{r}_z(k_f) = \begin{bmatrix} \delta \mathbf{x}_1(k_f) \\ \mathbf{0}_n \end{bmatrix} + \mathbf{B}_d^*(\mathbf{z}(k_f)) \left(\begin{bmatrix} \mathbf{0}_n \\ \frac{\delta \mathbf{x}_1(k_f)}{T} \end{bmatrix} + \boldsymbol{\Delta} \boldsymbol{\eta}_z(k_f - 1) \right). \quad (35)$$

It turns out that such an induced signature ($\frac{\delta \mathbf{x}_1(k)}{T}$) is larger than $\delta \mathbf{x}_1(k)$ when $T < 1$, and thus the signatures due to sensor failures results become even more obvious.

In order to improve the observer robustness, one could consider adopting the compensation of the disturbance term based on the current measurements from the system as in (28). However, this choice leads to an almost complete cancellation of fault signatures on the residuals. In fact, if $\widehat{\boldsymbol{\eta}}_d(k)$ is computed via (28) it is:

$$\widehat{\boldsymbol{\eta}}_d(k_f - 1) = \boldsymbol{\eta}_d(k_f - 1) + \begin{bmatrix} \delta \mathbf{x}_1(k_f) \\ \mathbf{0}_n \end{bmatrix}, \quad (36)$$

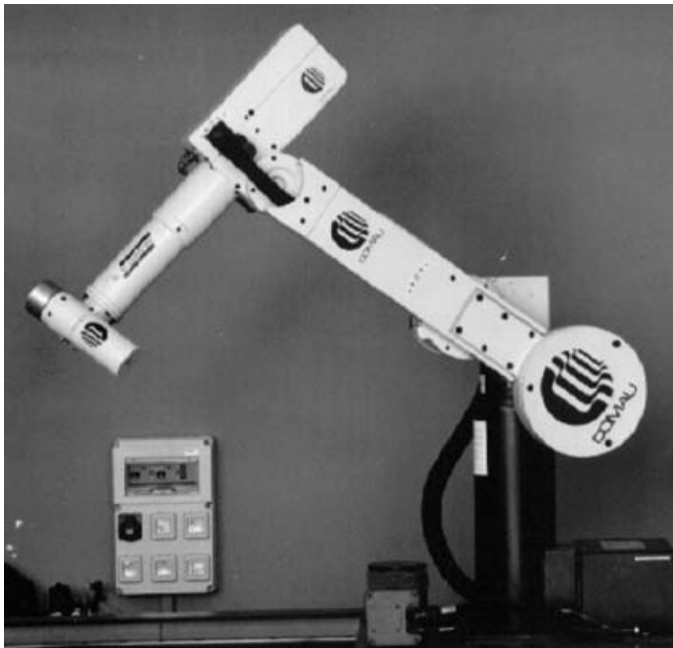


Fig. 2. The robot Comau Smart-3 S.

for a sensor fault, and

$$\hat{\eta}_d(k_f) = \eta_d(k_f) + \mathbf{B}_d(\mathbf{x}_1(k_f))\delta u(k_f), \quad (37)$$

in the case of an actuator fault. Hence, $\hat{\eta}_d$ compensates for the effect of the fault as well, which is seen as a disturbance, and thus the fault signature on the residuals becomes null.

IV. EXPERIMENTAL RESULTS

The FDI techniques described in the above are tested on the setup available in the laboratory. The robot is a Comau SMART-3 S industrial unit (Fig. 2). The manipulator has a six-revolute-joint anthropomorphic geometry with nonnull shoulder and elbow offsets and non-spherical wrist. The joints are actuated by brushless motors via gear trains; shaft absolute resolvers provide motor position measurements. The robot is controlled by the C3G 9000 control unit which has a VME-based architecture with 2 processing boards (Robot CPU and Servo CPU) both based on a Motorola 68020/68882. It is worth remarking that the SMART-3 S is a conventional industrial unit and not a research prototype; hence, all the typical drawbacks of industrial manipulators (e.g. joint friction, stiction and backlash due to the gear trains, disturbances on the torque delivered by the actuators, unmodeled elasticity of the joint shafts) are present.

An open version of the control unit has been developed which allows testing of advanced control algorithms on a conventional industrial robot. Communication between the VME bus of the C3G 9000 unit and the ISA bus of a standard PC is made possible by a bus-to-bus adapter board and a shared memory area available in the Robot CPU. Time synchronization is implemented by interrupt signals from the C3G to the PC with data exchange at a given sampling rate. A set of C routines are available to drive the bus adapter boards. At present, a PC Pentium is used as control unit.

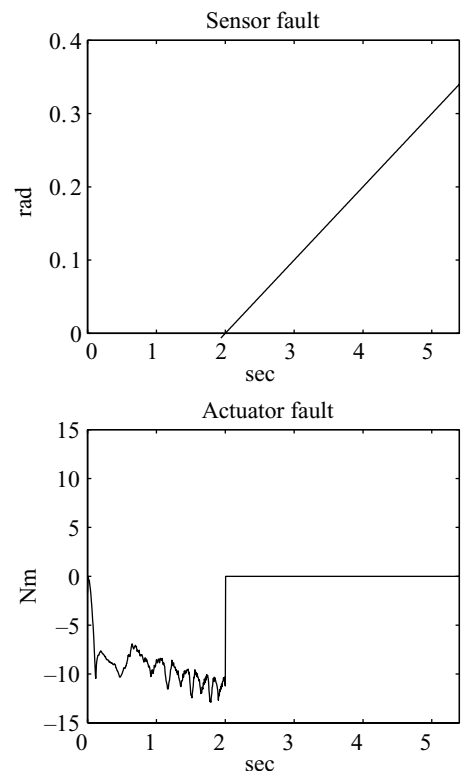


Fig. 3. Fault on sensor 5 measurements (top) – Fault on actuator 5 torque (bottom).

Various operational modes are available in the control unit, allowing the PC to interact with the original controller both at trajectory generation level and at joint control level. To implement model-based control schemes, the operational mode 4 is used in which the PC is in charge of computing the control algorithm and passing the references to the current servos through the communication link.

The dynamic model of the manipulator is expressed in terms of a minimum set of dynamic parameters, estimated on the basis of direct measurements on the manipulator.²⁴ In the experiments the manipulator is operated in closed loop: a joint-space inverse dynamics algorithm is adopted working at a sampling rate of 1000 Hz.

The diagnostic observer is implemented at a sampling rate of 1000 Hz ($T = 1$ ms), and the following gains have been adopted:

$$\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{I}_3. \quad (38)$$

The residuals are generated via (31). The nonlinear dynamics compensation term in the observer, \mathbf{h}_d , takes into account only the gravity terms in the model, and in the estimated inertia matrix used to compute \mathbf{B}_d only the diagonal terms have been kept.

In order to perform a proper fault detection and identification, suitably defined thresholds on the residuals components must be selected. This has been achieved by measuring the residuals obtained in a set of fault-free trajectories under various operating conditions. By inspecting the measured residuals and their deviation from zero, upper and lower threshold bounds have been set.

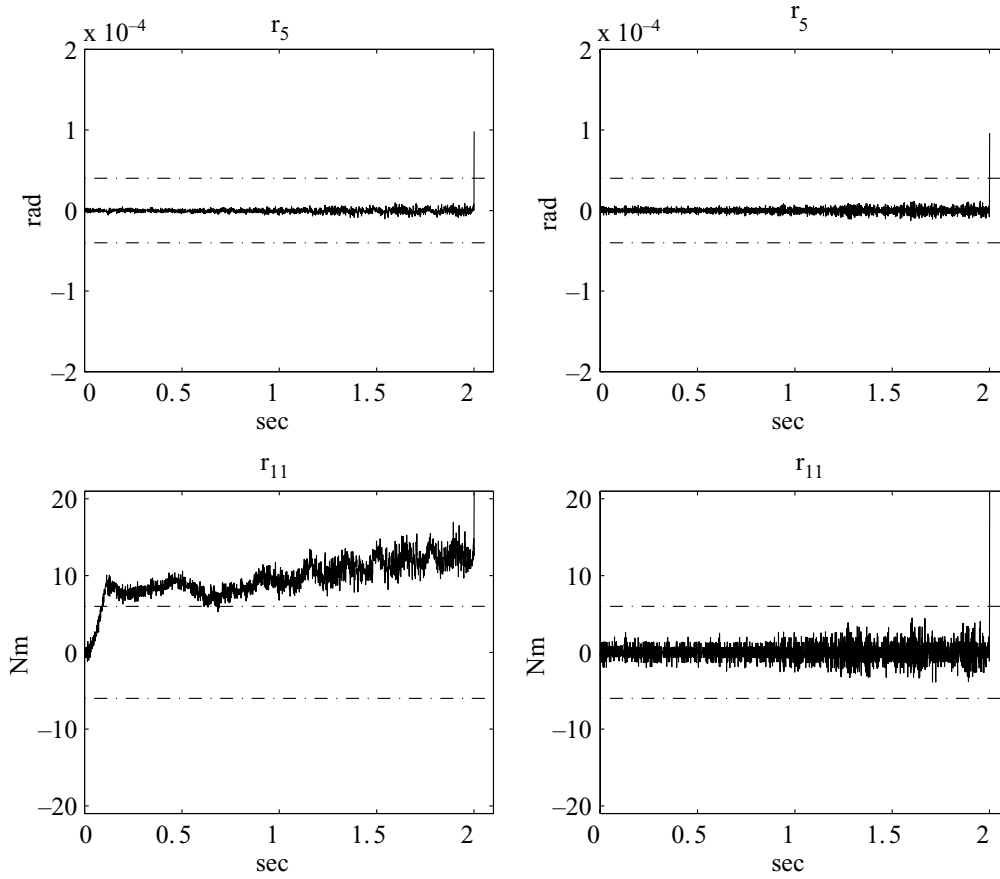


Fig. 4. Fault on sensor 5. Residuals for the 5-th joint. Algorithm without time-delayed compensation (left); algorithm with time-delayed compensation (right). Sampling time: $T = 1$ ms.

In the experiments a fifth-order polynomial trajectory is imposed at each joint with null initial and final velocities and accelerations; the initial joint configuration is

$$\mathbf{q}_0^T = [\pi/2 \quad -2\pi/3 \quad \pi/6 \quad -\pi/2 \quad \pi/2 \quad 0.0], \quad (39)$$

and the commanded joint displacement is given by:

$$\Delta \mathbf{q}^T = [\pi/4 \quad \pi/4 \quad -\pi/4 \quad 0.0 \quad -\pi/2 \quad \pi/2]. \quad (40)$$

The total (programmed) duration of the motion is 5 s.

In the first experiment a poorly detectable sensor fault is emulated: the sensor reading at the fifth joint drifts away from the true joint angle from the time $t_{fault} = 2.0$ s with a linear time law (i.e. $\delta q = 0.1t$). The time history of the fault is reported in Fig. 3. In the second experiment the same trajectory as above is commanded at each joint, but an actuator fault is emulated: the commanded torque at the fifth joint becomes null from the time $t_{fault} = 2.0$ s. The time history of the 5th joint torque with the emulated fault is reported in Fig. 3.

The results of the first experiment are reported in Fig. 4 in terms of the time histories of the two residuals corresponding to the 5th joint (i.e. r_5 and r_{11}). It can be recognized that, in the absence of time-delayed compensation of the disturbance term, the second component drifts away and overcomes the selected thresholds, thus generating a false alarm. On the other hand, a clear fault signature is obtained on both the

relevant components of the residual vector when the time-delayed compensation is adopted. As already pointed out (see (35)) the signature on the second component is due to the numerical reconstruction of the joint velocities. Remarkably, this feature is not to be regarded as a drawback; in fact, the effect of the failure on the residuals is amplified and thus becomes clearly detectable, even if the drifting behaviour of the failure is very smooth.

The results of the second experiments, reported in Fig. 5, confirm the potential of the proposed approach in the case of an actuator fault. In this case a clear signature can be observed on the second component of the residual, thus allowing the proper detection and isolation of the fault.

Finally, the observer has been implemented at a lower sampling rate, 200 Hz (i.e. $T = 5$ ms), and used to detect the same sensor fault considered above. Two cases have been compared: in the first case the time-delayed compensation is not performed, while in the second case the time-delayed compensation is kept. The results are reported in Fig. 6. Noticeably, in the first case a false alarm occurs, due to the drifting of the second component of the residual for the 5th joint, while in the second case the fault is properly detected.

V. CONCLUSION

In this paper an effective approach to fault detection and isolation for robot manipulators has been proposed, and the

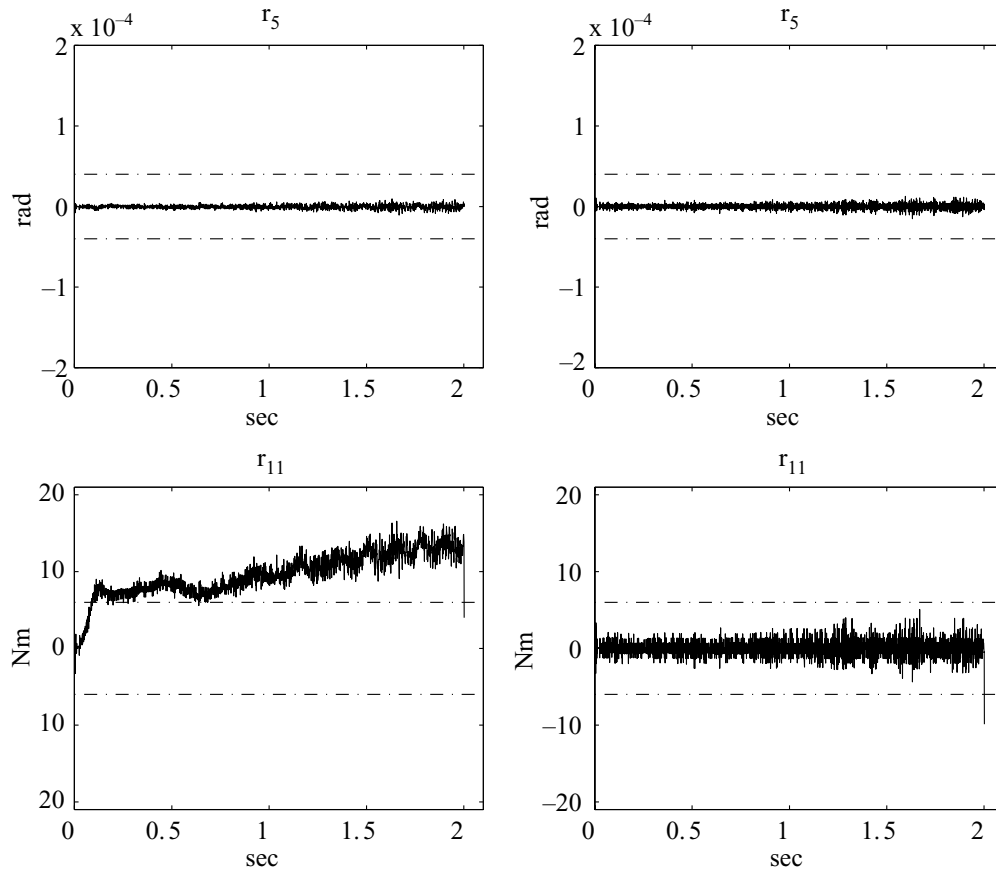


Fig. 5. Fault on actuator 5. Residuals for the 5-th joint. Algorithm without time-delayed compensation (left); algorithm with time-delayed compensation (right). Sampling time: $T = 1$ ms.

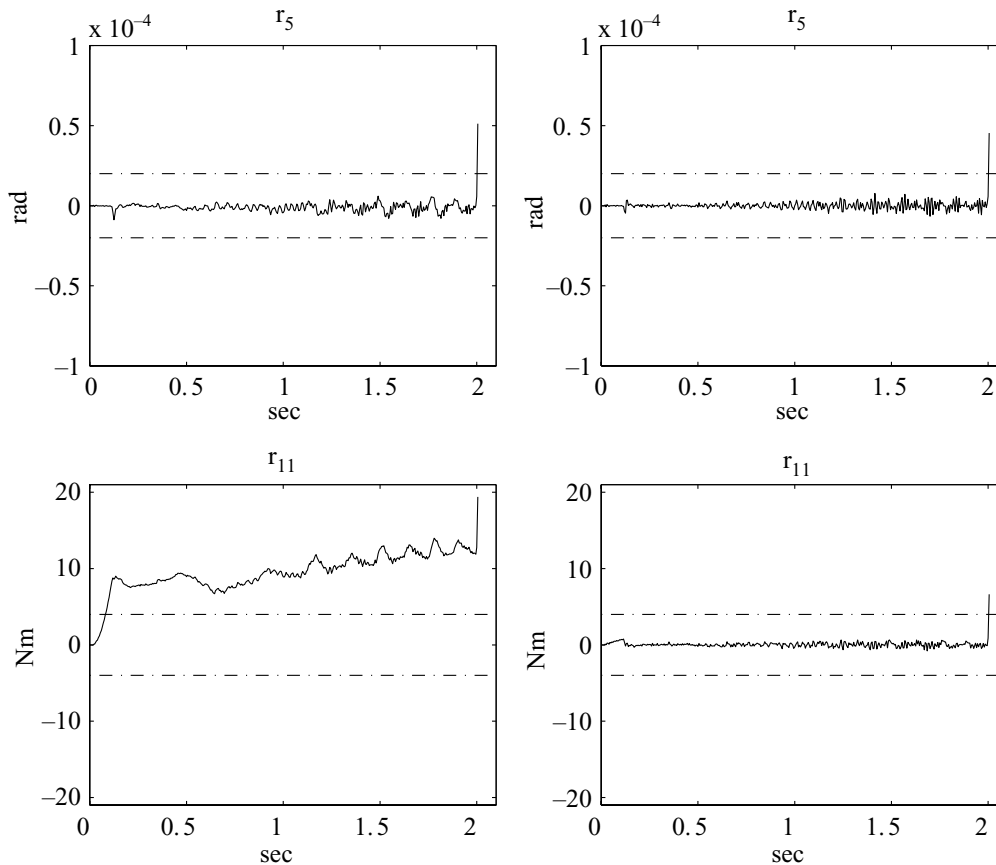


Fig. 6. Fault on sensor 5. Residuals for the 5-th joint. Algorithm without time-delayed compensation (left); algorithm with time-delayed compensation (right). Sampling time: $T = 5$ ms.

application to an industrial robot has been experimentally investigated. Namely, an observer-based technique for the detection of actuator and sensor faults has been developed and analyzed. Then, the effectiveness of the approach has been experimentally tested on an industrial robot with open control architecture. The results show that satisfactory fault detection and isolation is achieved even in the presence of partial model knowledge and measurement noise.

References

1. P. M. Frank, "Analytical and qualitative model-based fault diagnosis – A survey and some new results," *European Journal of Control* **2**, 6–28 (1996).
2. J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems* (Kluwer Academic Publisher, Boston, 1999).
3. R. Isermann and B. Freyermuth, "Process fault diagnosis based on process model knowledge – Part I: Principles for fault diagnosis with parameter estimation," *ASME J. of Dynamic Systems, Measurement, and Control* **113**, 620–626 (1991).
4. R. Isermann and B. Freyermuth, "Process fault diagnosis based on process model knowledge – Part II: Case study, experiments," *ASME J. of Dynamic Systems, Measurement, and Control* **113**, 627–633 (1991).
5. M. M. Polycarpou and A. J. Helmicki, "Automated fault detection and accomodation: A learning systems approach," *IEEE Transactions on Systems, Man, and Cybernetics* **25**, 1447–1458 (1995).
6. A. Vemuri and M. M. Polycarpou, "Robust nonlinear fault diagnosis in input–output systems," *International Journal of Control* **68**, 343–360 (1996).
7. M. A. Demetriou and M. M. Polycarpou, "Incipient fault diagnosis of dynamical systems using online approximators," *IEEE Transactions on Automatic Control* **43**, 1612–1617 (1998).
8. A. T. Vemuri, "Sensor Bias Fault Diagnosis in a Class of Nonlinear Systems," *IEEE Transactions on Automatic Control* **46**, 949–954 (2001).
9. X. Zhang, M. M. Polycarpou and T. Parisini, "A Robust Detection and Isolation Scheme for Abrupt and Incipient Faults in Nonlinear Systems," *IEEE Transactions on Automatic Control* **47**, 576–593 (2002).
10. B. Freyermuth, "An approach to model based fault diagnosis of industrial robots," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, Sacramento, CA (1991) pp. 1350–1356.
11. H. Schneider and P. M. Frank, "Observer-based supervision and fault detection in robots using nonlinear and fuzzy logic residual evaluation," *IEEE Transactions on Control Systems Technology* **4**, 274–282 (1996).
12. W. Dixon, I. Walker, D. Dawson and J. Hartranft, "Fault detection for robot manipulators with parametric uncertainty: a prediction-erro-based approach," *IEEE Transactions on Robotics and Automation* **16**, 689–699 (2000).
13. M. L. Visinsky, J. R. Cavallaro and I. D. Walker, "A dynamic fault tolerance framework for remote robots," *IEEE Transactions on Robotics and Automation* **11**, 477–490 (1995).
14. E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on Automatic Control* **29**, 603–614 (1984).
15. A. T. Vemuri, M. M. Polycarpou and S. A. Diakourtis, "Neural network based fault detection in robotic manipulators," *IEEE Transactions on Automatic Control* **14**, 342–348 (1998).
16. A. De Luca and R. Mattone, "Actuator failure detection and isolation using generalized momenta," *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*, Taipei, Taiwan (2003) pp. 634–639.
17. A. De Luca and R. Mattone, "An adapt-and-detect actuator FDI scheme for robot manipulators," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA (2004) pp. 4975–4980.
18. F. Caccavale and I. D. Walker, "Observer-based fault detection for robot manipulators," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM (1997) pp. 2881–2887.
19. F. Caccavale, "Experiments of Observer-based Fault Detection for an Industrial Robot," *Proceedings of the 1998 IEEE International Conference on Control Applications*, Trieste, Italy (1998) pp. 480–484.
20. G. de Vahl Davis, *Numerical Methods in Engineering and Science* (Allen & Unwin, London, 1986).
21. C. P. Neumann and V. D. Tourassis, "Discrete dynamic robot models," *IEEE Transactions on Systems, Man, and Cybernetics* **15**, No. 2, 193–204 (1985).
22. K. Youcef-Toumi and O. Ito, "A time delay controller design for systems with unknown dynamics," *ASME Journal of Dynamic Sytems, Measurement, and Control* **112**, No. 1, 133–142 (1990).
23. R. H. C. Takahashi and P. L. D. Peres, "Unknown input observers for uncertain systems: A unifying approach," *European Journal of Control* **5**, 261–275 (1999).
24. G. Antonelli, F. Caccavale and P. Chiacchio, "A systematic procedure for the identification of dynamic parameters of robot manipulators," *Robotica* **17**, part 3, 427–435 (1999).